TP Génie logiciel – Scrum

Sprint 3 :

Réunion de planification de sprint :

Durant ce sprint, on réalise un code qui peut convertir des fichiers **PDF** aux fichiers **XML** en utilisant le langage Ruby.

On aura comme sortie un fichier XML qui suit la structure suivante :

<article>

<preamble> Le nom du fichier d'origine </preamble>

<titre> Le titre du papier </titre>

<auteur> La section auteurs et leur adresse </auteur>

<abstract> Le résumé de l'article </abstract>

<biblio> Les références bibliographiques du papier</biblio>

</article>

Pour pouvoir réaliser ce travail on s'est bien organisé par répartir les taches(les recherches, le développement, la documentation).

Daily Meetings :

Durant ces meetings (3 fois), on s'est mets au courant de notre avancement et les problèmes rencontrés.

Ils durés pas plus du 15 min, mais ca nous a permets de visualiser la progression de ce travail.

Revue de sprint :

A la fin de ce sprint, on a pu faire ce travail par ce code expliqué ci-dessous :

Pour la conversion on a utilisé ‘pdf-Reader’ qui permet de lire un fichier pdf

Et le convertir

```
require 'pdf/reader'

# Starting time execution
start = Process.clock_gettime(Process::CLOCK_MONOTONIC)

ARGV.each do |filename|
        # Start converting using PDF-Reader
        PDF::Reader.open(filename) do |reader|

                puts "Converting : #{filename}"
                pageno = 0
                txt = reader.pages.map do |page|

                pageno += 1
                begin
                        print "Converting Page #{pageno}/#{reader.page_count}\r"
                        page.text
                rescue
                        puts "Page #{pageno}/#{reader.page_count} Failed to convert"
                        ''
                end

        end
```

Après la lecture de pdf, on extrait les informations nécessaires qu’on veut afficher dans les balises spécifié.

En commençant par abstraire le titre du fichier pdf en utilisant une boucle qui

Récupère la ligne d’une longueur moins de 40 et on sort de la boucle c’est le numéro de ligne est null.

```
title = ""
paragraph = ""
paragraphs = []
reader.pages.each do |page|
        lines = page.text.scan(/^.+/)
        x = 0
        # Title
        lines.each do |line|
                if line.length < 40
                        title += " #{line}"
                        if lines.index(line) == 0
                                break
                        end
                end
        end
```

Ensuite on récupère la partie abstract du pdf

```
        # Abstract
        lines.each do |line|
                if line.length > 40
                        paragraph += " #{line}"
                        paragraphs << paragraph
                        if lines.index(line) == 10 # if abstract has 10 lines
                                break
                        end
                end
                paragraph = ""
        end
        break
end
```

Et à la fin on fait l'insertion au fichier xml par le code suivant ;

File.write filename+'.xml', "<article>\n<preamble>" + filename + "</preamble>\n<titre>" + title + "</titre>\n<abstract>" + paragraphs.join("") + "</abstract>\n</article>"

Le fichier XML sotie est à la forme suivante :

<article>

<preamble> Le nom du fichier d'origine </preamble>

<titre> Le titre du papier </titre>

<abstract> Le résumé de l'article </abstract>

</article>

-exemple :

```
<article>
<preamble>Das_Martins.pdf</preamble>
<titre> 1    Introduction     Summaries should be short.</titre>
<abstract>         A Survey on Automatic Text Summarization                    Dipanjan
Das          Andr F.T. Martins                          Language Technologies
Institute                          Carnegie Mellon University
fdipanjan, afmg@cs.cmu.edu                                  November 21,
2007                                  Abstract         The increasing availability of
online information has necessitated intensive      research in the area of automatic text
summarization within the Natural Lan-      guage Processing (NLP) community. Over the past
half a century, the prob-      lem has been addressed from many di erent perspectives, in
varying domains</abstract>
</article>
```

***Les points à améliorer :**

-prévenir plus de meetings entre tous les membres d'équipe.

-respecter la boite de temps du chaque évènement.

***Document supplémentaires** :

Ce qu'on a fait (1ᵉʳ)

pdftotxt fichier . pdf

↳ fichier . txt .

(2) - fichier . txt | Titre .
Auteur .
Résumé

(3) - fichier . XML

On nous demande d'afficher des balise

<article>
<préambules> nom fichier .
<auteurs> .
<abstracts> .
<biblio> . Référence bibliographique .
</article> .

Saad pense à une manière de toutes les résumes .
Mélissa bibliothèque [illegible] + Partie rapport .
Mohamed pense à la biblio .
Hamed . transformer en XML .

idée :
commencer d'abord le pdf en fichier . txt avec pdftotext

- Extraire les information .

- write to file XML .

le nom de l'article : On le récupère avec ARVEn [illegible] .
auteur ?
abstract ?
biblio ?